Credit Card Default Prediction Using Machine Learning

CHANG Yung-Hsuan

eiken.sc11@nycu.edu.tw

July 19, 2025

Abstract

This project applies supervised machine learning to detect credit card default on a real-world dataset of 30,000 Taiwanese clients, where only ~22% defaulted. Five models—logistic regression (LR), decision tree (DT), *k*-nearest neighbors (*k*-NN), random forest (RF), and XGBoost (XGB)—were evaluated under a common, imbalance-aware pipeline (`class_weight='balanced'` for LR/DT/RF and `scale_pos_weight` for XGB, all tuned for default-class F1 via cross-validation). Despite high headline accuracy (e.g., *k*-NN at ~81%), minority-class recall was the binding constraint. Once imbalance handling was equalized, the weighted models clustered around F1 ~0.50–0.54 with recall ~0.60–0.64, and their differences fell within bootstrap confidence intervals; *k*-NN, which cannot be class-weighted, trailed. The project highlighted the challenges of rare-event prediction and the trade-offs between recall and false positives. The author[1] led the modeling and analysis.

[1]This report originated as a semester-long group project at École Polytechnique with Carla GUINEA CARRANZA. The present version, however, is independent work of my own: all data preprocessing, modeling, analysis, the imbalance-aware re-evaluation reported here, and the write-up were carried out solely by me, with the original project serving only as the starting point.

Table of Contents

List of Tables

List of Figures

1 Introduction

Credit card default prediction has become a critical challenge in finance, as ineffective models can incur major financial and reputational losses. While machine learning offers potential, rare-event classification remains difficult due to severe class imbalance: high accuracy may hide poor detection of default cases, the so-called "accuracy paradox."

This report investigates five supervised learning models for detecting credit card defaults using the Default of Credit Card Clients dataset from the UCI Machine Learning Repository (Yeh 2009). The dataset includes 30,000 Taiwanese credit card accounts with 23 features, and ~22% labeled as defaulting. It captures realistic credit risk scenarios and has been widely cited in the literature (Yeh and Lien 2009; Bhandary and Ghosh 2025).

Our goal was to compare five classifiers—logistic regression (LR), decision tree (DT), k-nearest neighbors (k-NN), random forest (RF), and XGBoost (XGB)—under a consistent, imbalance-aware pipeline—uniform class weighting wherever the algorithm supports it, and a single default-class F1 tuning objective. We emphasized minority-class performance, evaluating models with F1-score, precision, recall, and ROC-AUC instead of overall accuracy.

2 Methodology

2.1 Data Preprocessing

The Default of Credit Card Clients dataset (Yeh 2009) contains 30,000 records of Taiwanese credit card holders. Each record includes demographic variables (age, gender, education, marital status), financial indicators (credit limit, bill and payment amounts), and six months of payment status history (April–September 2005). The binary target variable indicates whether the client defaulted in the following month. The data are highly imbalanced, with only 22.12% defaults, meaning that a naive classifier predicting "no default" would achieve 77.88% accuracy, a misleading benchmark.

The data preparation pipeline included the following steps:

(a) **Data Cleaning**. Removed the ID column; recoded categorical variables (`SEX`) to binary for clarity.

(b) **Feature Engineering**. Constructed summary features such as `MAX_DELAY` and `AVG_DELAY` from the six `PAY_n` variables, and computed `PAYMENT_DIFF_1` to `PAYMENT_DIFF_6` to capture bill–payment gaps that may signal underpayment.

(c) **One-Hot Encoding**. One-hot encoded the categorical fields—education level (`EDUCATION`), marital status (`MARRIAGE`), and the monthly payment status (`PAY_0`–`PAY_6`)—to handle these categories without imposing an arbitrary ordinal relationship. This expanded the feature space accordingly.

(d) **Train-Test Split**. Stratified 80/20 split preserving the default ratio; all subsequent fitting (skew selection, the signed-log columns, and scaling) is derived from the **training** set only, with the test set held out.

(e) **Skewness Reduction**. Monetary features (`BILL_AMT1`–`6`, `PAY_AMT1`–`6`, and the engineered `PAYMENT_DIFF` gaps) are heavy-tailed, and bill amounts can be negative (credit balances or overpayment). On the training set we flagged features with absolute skewness above 1 (Bulmer's rule of thumb for high skew)—which selected all monetary features—and applied a signed logarithmic transform $f(x) = \text{sign}(x) \cdot \log(1 + |x|)$. The signed form compresses both tails, handles negatives natively without an artificial shift, and is computed element-wise, so it introduces no train–test leakage. Figure 1 and Figure 2 show a typical bill-amount feature before and after the transform.

(f) **Feature Scaling**. Min-max scaling of continuous variables, fit on the training set only, for scale-sensitive models such as k-NN and LR.

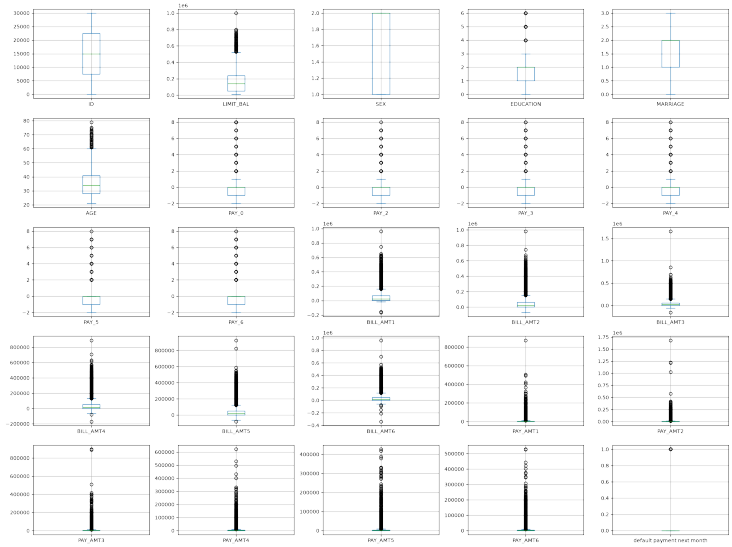

Figure 1: Feature distributions before the signed-log transform, where the monetary amounts are heavily right-skewed and span several orders of magnitude.

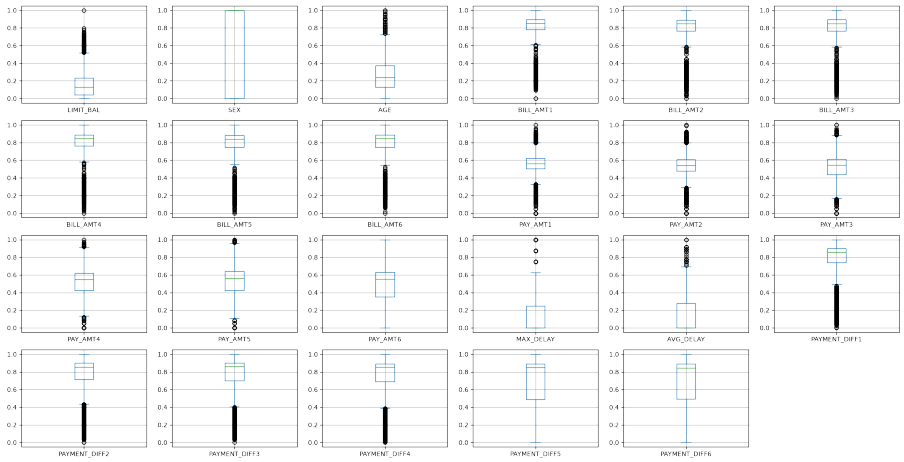

Figure 2: Training-set feature distributions after feature engineering, the signed-log transform, and min-max scaling, with the heavy tails compressed and the feature scales made comparable.

2.2 Modeling and Hyperparameter Tuning

We evaluated five supervised learning models—LR, DT, *k*-NN, RF, and XGB—chosen for their methodological diversity. These span linear models, tree-based classifiers, instance-based methods, and ensemble learners, offering a broad baseline for imbalanced classification.

To address class imbalance consistently, we applied built-in cost weighting to every model that supports it: `class_weight='balanced'` for LR, DT, and RF, and the equivalent `scale_pos_weight` = $n_{\text{neg}}/n_{\text{pos}}$ (computed on the training set, ≈ 3.52 here) for XGB. k-NN has no built-in class-weighting mechanism in scikit-learn, so it is the one model trained without imbalance correction—a caveat to its low recall rather than evidence of an inferior algorithm. No oversampling (e.g., SMOTE) was used in primary training; it is discussed later as an enhancement.

Each model underwent grid search with 3-fold stratified cross-validation on the training set, all optimizing the same objective—F1-score for the default class—so that tuning was comparable across models. Hyperparameter ranges included regularization strength (LR), tree depth and split size (DT, RF), neighbor count (k-NN), and boosting parameters (XGB). Final models were selected based on best CV F1-score.

Given that a naive model predicting all non-defaults would achieve ~78% accuracy, we focused on metrics that reflect positive-class performance: Recall (true default detection), Precision (accuracy of positive predictions), F1-score (balance of recall and precision), and ROC-AUC (ranking ability). Accuracy was reported but interpreted cautiously due to its sensitivity to class imbalance.

To detect overfitting, we compared training and test metrics; large gaps indicated potential generalization issues. Final test results are presented next, with corresponding observations on model robustness.

3 Results

After training and tuning, each model was evaluated on the test set (20% of the data not seen during training). Table 1 summarizes the performance of all five models on the test data across the relevant metrics:

Table 1: Test-set performance of the five default-prediction models. Accuracy is overall, whereas precision, recall, and F1-score are reported for the positive class "default = 1". The 95% bootstrap confidence intervals ($B = 2000$) for the default-class F1-score are LR: $[0.500, 0.542]$, DT: $[0.479, 0.520]$, k-NN: $[0.425, 0.478]$, XGB: $[0.505, 0.547]$, and RF: $[0.520, 0.563]$; the corresponding recall intervals are LR: $[0.576, 0.628]$, XGB: $[0.610, 0.662]$, and RF: $[0.595, 0.647]$.

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
LR	0.7552	0.4592	0.6021	0.5210	0.7651
DT	0.7257	0.4188	0.6202	0.5000	0.7429
k-NN	0.8107	0.6275	0.3542	0.4528	0.7431
XGB	0.7465	0.4485	0.6360	0.5260	0.7766
RF	0.7680	0.4810	0.6209	0.5421	0.7749

With imbalance handling equalized, the four weighted models (LR, DT, RF, XGB) reach broadly similar default-class F1 (~0.50–0.54) and recall (~0.60–0.64); XGB attains the highest recall (0.636) and RF the highest F1 (0.542), but the gaps among them are comparable to their bootstrap confidence intervals, so we read this as a cluster rather than a strict ranking. This is consistent with prior comparisons on this dataset, where non-linear methods perform at least as well as linear ones Yeh and Lien (2009).

The persistent pattern is the recall–precision trade-off: lifting recall to ~0.6 pushes precision down to ~0.42–0.48. k-NN is the exception—without class weighting it stays at recall ~0.35 and the lowest F1 (0.453). ROC-AUC, which is threshold-independent, separates the models far less than the thresholded metrics (all five fall within 0.74–0.78), consistent with most of the apparent differences coming from operating-point and weighting effects rather than ranking ability.

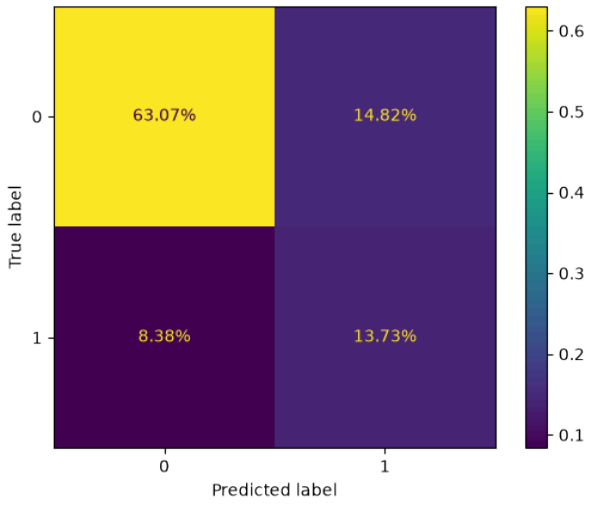

Figure 3: Test-set confusion matrix for the random forest, with each cell expressed as a percentage of all test cases.

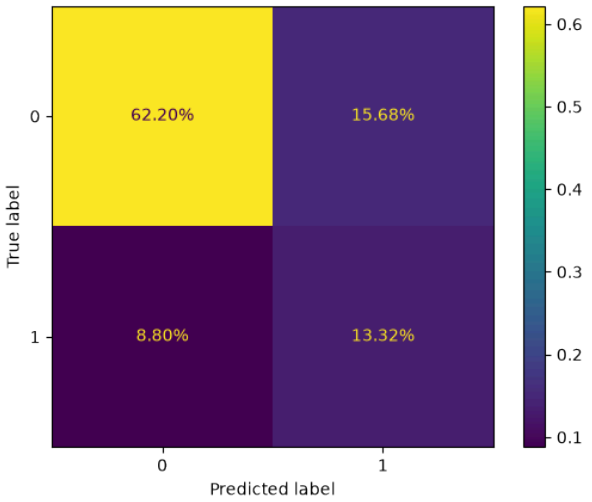

Figure 4: Test-set confusion matrix for the weighted logistic regression, with each cell expressed as a percentage of all test cases.

To assess overfitting, we compared training and test scores. The weighted LR and DT showed small gaps (train vs. test F1 within ~0.01–0.02), indicating good generalization. RF and XGB had moderately higher training F1 (~0.57–0.60 vs. test ~0.53–0.54), expected given their flexibility. *k*-NN showed severe overfitting (training F1 ~0.99 vs. test ~0.45): with distance weighting each training point is its own zero-distance neighbor, so training F1 approaches 1.0 regardless of *k*—it effectively memorizes the training data, especially problematic under imbalance.

Confusion matrices further reveal model behavior. With class weighting, LR, DT, RF, and XGB all catch a comparable share of defaults (recall ~0.60–0.64) at the cost of more false positives, whereas the unweighted *k*-NN still mostly predicts the majority class and misses most defaults. Figure 3 and Figure 4 illustrate this: the weighted RF and LR now reach ~60% recall on essentially the same confusion structure, underscoring that LR's earlier low recall was a weighting artifact rather than a modeling limit.

Lastly, the ROC-AUC values (Figure 5) summarize model ranking quality across thresholds. XGB had the highest ROC-AUC (0.777), followed closely by RF (0.775) and the weighted LR (0.765), while DT and *k*-NN trailed (~0.743). The narrow AUC spread—against the wide spread in thresholded recall

—indicates that most of the apparent differences reflect operating-point and weighting choices rather than ranking ability.

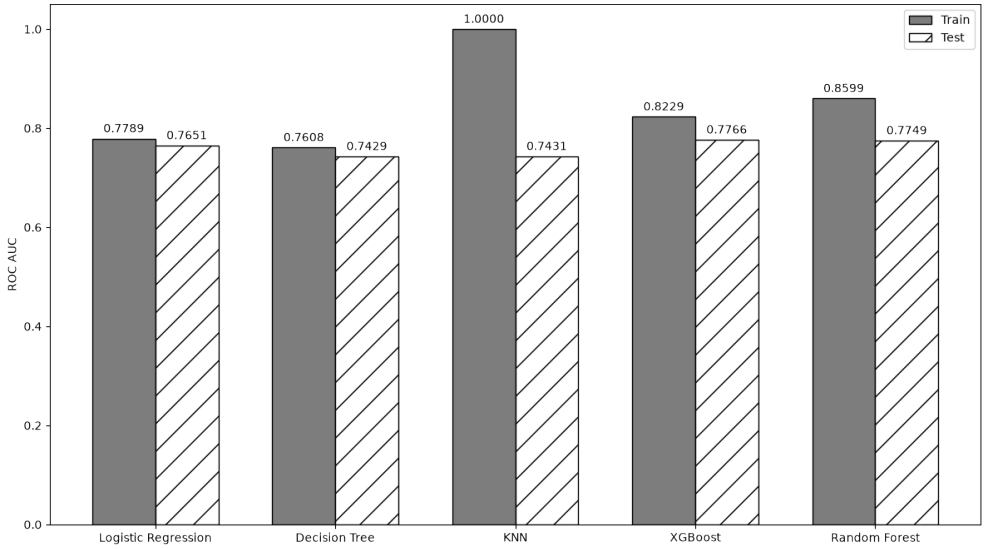

Figure 5: Training- and test-set ROC-AUC for the five models, whose values cluster far more tightly than the thresholded metrics do.

4 Discussion

Our findings highlight the challenge of imbalanced classification in credit default detection. While most models achieved ~80% accuracy, beating the 77.9% baseline, this masked weak minority-class performance. Models generally struggled to achieve both high recall and precision for defaults, reflecting the real-world trade-off: failing to detect defaults (false negatives) is costly, but excessive false alarms (false positives) burdens operations.

Among the weighted models the differences were small. RF posted the highest F1 (0.542), and XGB the best AUC (0.777) and highest recall (0.636), with a properly weighted LR close behind on F1 (0.521); the standalone DT was competitive on recall but had the lowest precision, while generalizing well. Because all four operate near the same recall–precision frontier, the practical lever is the decision threshold: institutions can move along that frontier—trading precision for recall or vice versa—rather than expecting any one model to dominate.

Once weighted, LR recovers recall to ~0.60 and F1 to ~0.52, comparable to the tree ensembles; the earlier impression of LR as a weak linear model was largely an artifact of it being unweighted. Any residual gap to the best model is small (~0.02 in F1), with overlapping confidence intervals. k-NN fared worst on recall, both because it cannot be class-weighted in scikit-learn and because of the curse of dimensionality.

We addressed class imbalance through class weighting and F1-score-focused evaluation but did not apply SMOTE or threshold tuning in baseline models. These remain viable enhancements: oversampling could increase recall, and lowering classification thresholds could align detection with operational cost-risk tradeoffs (Chawla et al. 2002). Because all metrics come from a single 80/20 split, we computed bootstrap 95% CIs ($B = 2000$) on the test set; the F1 differences among the weighted models are comparable to the CI width (~±0.02), so we avoid over-interpreting their ranking.

Overfitting control was essential. Unpruned trees and a near-memorizing k-NN showed inflated training scores but weak test performance; for k-NN this is driven by distance weighting (each training point is its own zero-distance neighbor, so training F1 approaches 1.0 regardless of k)—not by large k. Grid search with cross-validation helped identify model complexity that generalizes; the cross-validated curves show diminishing returns as depth grows (Figure 6 and Figure 7).

Limitations remain. Our models used static data and did not model temporal trends, which could improve prediction (e.g., via RNNs). Most models are also black boxes; applying SHAP for interpretability could aid model trust and insight. Our ensemble results (AUC ~0.77, F1-score ~0.55) align with those reported by Bhandary and Ghosh (2025). Even well-tuned models miss a sizable portion of defaults, suggesting room for further improvement via stacking or cost-sensitive learning.

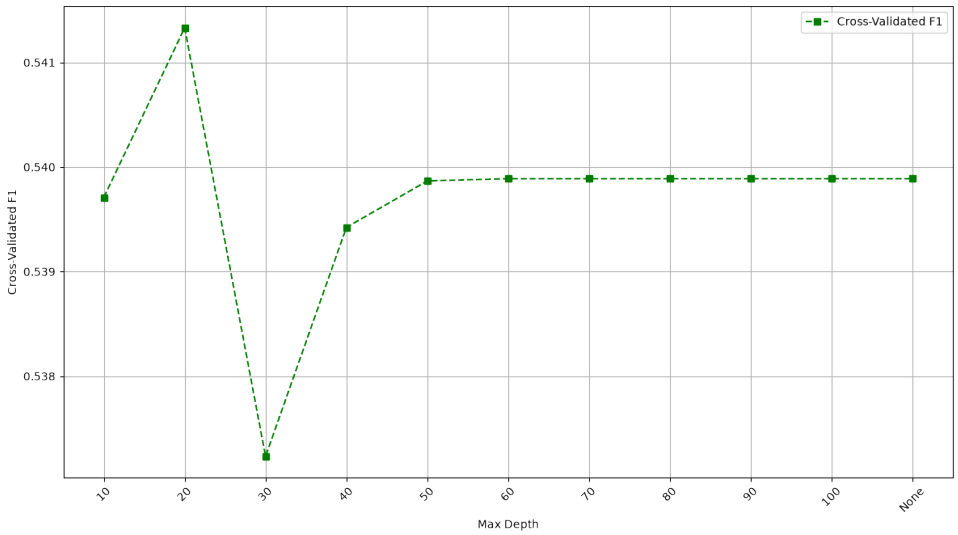

Figure 6: Cross-validated default-class F1-score of the random forest as a function of `max_depth` (3-fold cross-validation on the training set), which plateaus once the trees are deep enough.

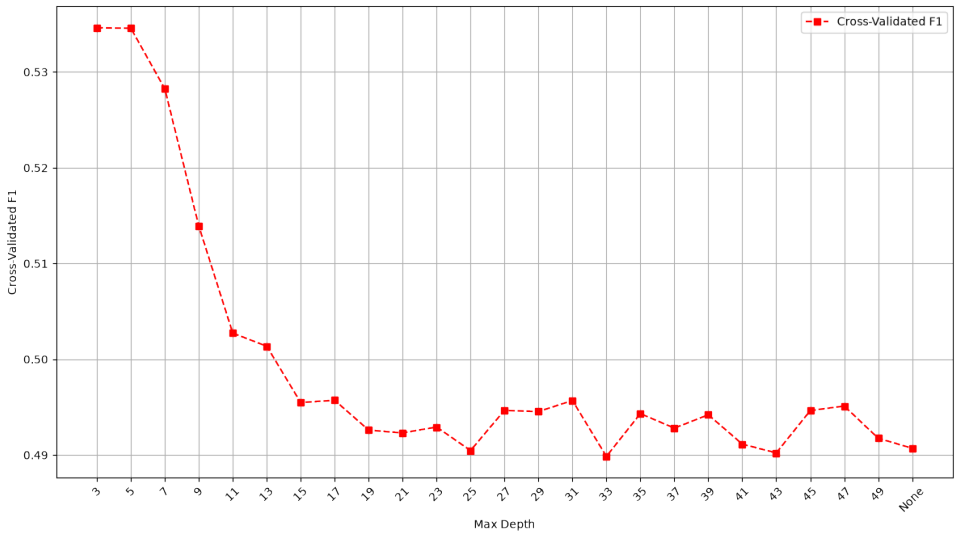

Figure 7: Cross-validated default-class F1-score of XGBoost as a function of `max_depth` (3-fold cross-validation on the training set), which peaks at shallow depth and declines as depth grows.

5 Conclusion

This project explored credit card default prediction using supervised machine learning on an imbalanced dataset. We evaluated five models—LR, DT, *k*-NN, RF, and XGB—within a structured pipeline involving feature engineering and hyperparameter tuning. Once imbalance handling was equalized, tree ensembles and a properly weighted logistic regression performed comparably; no single model clearly

dominated, and the recall–precision trade-off was the dominant theme. All models faced the inherent trade-off of rare-event detection: improving recall often lowered precision, and vice versa.

The author led data preprocessing, feature design, pipeline implementation, and performance analysis. This hands-on work emphasized the value of domain-informed features (e.g., delay patterns), evaluation beyond accuracy, and strategies for imbalanced classification.

Our best model identified ~60% of defaulters at ~48% precision—better than chance, but far from ideal, with the weighted models clustered within overlapping confidence intervals. Future improvements may include SMOTE oversampling (Chawla et al. 2002), threshold tuning, or stacking. Incorporating temporal patterns and tools like SHAP could also enhance interpretability and risk insights.

In conclusion, the project illustrates both the promise and limits of machine learning in financial risk detection. Ongoing progress depends on combining technical refinement with domain awareness and cost-sensitive goals.

Bibliography

Bhandary, R., and B. K. Ghosh. 2025. "Credit Card Default Prediction: An Empirical Analysis on Predictive Performance Using Statistical and Machine Learning Methods." *Journal of Risk and Financial Management* 18 (1): 23. https://doi.org/10.3390/jrfm18010023.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. "SMOTE: Synthetic Minority over-Sampling Technique." *Journal of Artificial Intelligence Research* 16 : 321–57. https://doi.org/10.1613/jair.953.

Yeh, I.-C. 2009. "Default of Credit Card Clients." UCI Machine Learning Repository,. https://doi.org/10.24432/C55S3H.

Yeh, I.-C., and C.-H. Lien. 2009. "The Comparisons of Data Mining Techniques for the Predictive Accuracy of Probability of Default of Credit Card Clients." *Expert Systems with Applications* 36 (2): 2473–80. https://doi.org/10.1016/j.eswa.2007.12.020.